Knuth-Bendix procedure and non deterministic behavior. An example

Isabelle Gnaedig

INRIA

UNITÉ DE RECHERCHE
INRIA-LORRAINE

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France

Tél. (1) 39 63 55 11

Rapports de Recherche

N° 733

KNUTH-BENDIX PROCEDURE AND NON DETERMINISTIC BEHAVIOR - AN EXAMPLE-

I. GNAEDIG

OCTOBRE 1987

Knuth-Bendix procedure and non deterministic behavior -An example-

*

Procédure de Knuth-Bendix et indéterminisme -Un exemple-

I. Gnaedig
INRIA
Campus Scientifique BP 239
54506 Vandoeuvre

Abstract

In this note, we present and study a typical example to illustrate the non determinism of the Knuth-Bendix algorithm with respect to the ordering it uses. Our example shows how different orderings on terms lead to different completion processes, and how eleven different completion sessions produce only two different rewrite systems.

Résumé

Dans cette note, nous présentons un exemple typique, illustrant le non-déterminisme de l'algorithme de Knuth-Bendix vis à vis de l'ordre qu'il utilise. Cet exemple montre comment différents ordres sur les termes conduisent à différents processus de complétion et comment onze complétions différentes peuvent ne produire que deux systèmes de réécriture distincts.

Starting from a set of equations, the Knuth-Bendix (KB in short) completion procedure computes an equivalent term rewriting system, which has the same deduction power than the initial set. Thus the equational deduction in the initial set of equations can be replaced by the reduction process with the corresponding term rewriting system [5].

To ensure this equivalence, the resulting set of rules, if it exists, has to satisfy the two essential properties of local confluence and termination. The local confluence is ensured by computing critical pairs between the rules of the current system; the termination is ensured by orienting equations into rules l→r in order to have $l > r$ for a given reduction ordering $>$ [1].

Let us explain how such an ordering works in the completion procedure. The usual orderings used in the implementation of the completion procedure (especially in the rewriting laboratory REVE [3]) are the RDO and the RPO. Both orderings are based themselves on an ordering on operators called precedence. When the completion procedure starts with a set of equations, the precedence is empty. There are only the following variable constraints, to ensure that the given ordering is a reduction ordering:

$$\textit{for every variable } x, y, \textit{ we have } x \neq y$$
$$\textit{for every operator } f, \textit{ we have } x \neq f.$$

The precedence is incrementally increased during the completion, whenever the current precedence fails to orient an equation. In REVE, suggestions for a precedence, based on an RDO computation, are proposed to the user, who can choose among them [4].

It is well known that the set of rules R computed by the completion procedure is unique, when it exists, for an input set of equations and a reduction ordering $>$ (for a study of existence and construction of term rewriting systems, see [2]). So, proposing different precedence possibilities in the completion leads to different orderings, therefore to different possibly convergent sets of rules R. This problem was pointed out in [6] when finding a new complete specification for groups, and in [7].

On another hand, ordering rules differently can lead to different completion behaviors (generating different rules), although it yelds the same final system. We present here a new example to illustrate the non determinism of the KB procedure, when starting with an empty precedence. This example was observed by the author, when studying the completion problems in order sorted algebras. The initial set of rules was found in [8], and then

completed by running the KB algorithm implemented in REVE. It describes axioms of the boolean algebra:

$$not(f(x)) = f(x) \quad (1)$$
$$y + y = y \quad (2)$$
$$y \,\&\, y = y \quad (3)$$
$$y + not(y) = 1 \quad (4)$$
$$not(y) + y = 1 \quad (5)$$
$$y \,\&\, not(y) = 0 \quad (6)$$
$$not(y) \,\&\, y = 0 \quad (7)$$
$$not(0) = 1 \quad (8)$$
$$not(1) = 0 \quad (9)$$

where *not* is the unary logical negation operator, + the exclusive disjunction and f any unary operator. Axiom (1) can be a theorem to be refuted.

Let us look at a first completion case. The equations $1, 2, 3$ are oriented without precedence hypothesis, respectively in:

$$not(f(x)) \rightarrow f(x)$$
$$y + y \rightarrow y$$
$$y \,\&\, y \rightarrow y.$$

Then the critical pair $not(0) = 1$ is generated and three possibilities are proposed to orient it:

$$0 > 1 \quad (1)$$
$$not > 1 \quad (2)$$
$$1 > 0, 1 > not. \quad (3)$$

Let us choose the first hypothesis. The equation is then oriented in

$$not(0) \rightarrow 1.$$

The new critical pair $not(1) = 0$ is then generated and the following precedence cases are presented:

$$not > 0 \quad (1)$$
$$0 > not. \quad (2)$$

Let us choose the first hypothesis again. The equation is oriented in

$$not(1) \to 0.$$

Thus the equations $4, 5, 6, 7$ are automatically oriented in:

$$\begin{aligned} y + not(y) &\to 1 \\ not(y) + y &\to 1 \\ y \,\&\, not(y) &\to 0 \\ not(y) \,\&\, y &\to 0. \end{aligned}$$

Then the rules

$$\begin{aligned} 1 \,\&\, 0 &\to 0 \\ 0 \,\&\, 1 &\to 0 \\ 1 + 0 &\to 1 \\ 0 + 1 &\to 1 \end{aligned}$$

are generated. The equation $f(x) = 1$ appears and is oriented in

$$f(x) \to 1$$

if the precedence proposition $f > 1$ is chosen. The rule

$$0 \to 1$$

is added and after reduction of the rules, the complete system

$$\begin{aligned} y + y &\to y && (1) \\ y \,\&\, y &\to y && (2) \\ not(1) &\to 1 && (3) \\ y + not(y) &\to 1 && (4) \\ not(y) + y &\to 1 && (5) \\ y \,\&\, not(y) &\to 1 && (6) \\ not(y) \,\&\, y &\to 1 && (7) \\ f(x) &\to 1 && (8) \\ 0 &\to 1 && (9) \end{aligned}$$

called A, is obtained.

Let us look at another possibility. Start the completion as previously, but at the first choice, valid the second possibility instead of the first one:

$$not > 1.$$

As in the first case, the equation $not(0) = 1$ is ordered into:

$$not(0) \rightarrow 1.$$

From here, the different precedence we choose leads to different computations. Now, the equations 4 and 5 are ordered into

$$y + not(y) \rightarrow 1$$
$$not(y) + y \rightarrow 1$$

without further hypothesis of precedence. The rules

$$0 + 1 \rightarrow 1$$
$$1 + 0 \rightarrow 1$$

are generated, and the equation $not(1) = 0$ appears, which cannot be oriented with the current precedence. Completing it, the following are proposed:

$$1 > 0 \tag{1}$$
$$not > 0 \tag{2}$$
$$0 > not. \tag{3}$$

With the first possibility, the equation is oriented into

$$not(1) \rightarrow 0.$$

Then the equations 6 and 7 are oriented into

$$y \,\&\, not(y) \rightarrow 0$$
$$not(y) \,\&\, y \rightarrow 0.$$

The rules

$$1 \,\&\, 0 \to 0$$
$$0 \,\&\, 1 \to 0$$

are generated; the new equation $f(x) = 1$ is oriented into

$$f(x) \to 1$$

in accepting the unique precedence case proposed: $f > 1$. The equation

$$1 \to 0$$

is then generated (whereas 0→1 is generated in the first completion).

The resulting system B is symmetrical to the previous one, with respect to the constants 0 and 1:

$$y + y \to y \quad (1)$$
$$y \,\&\, y \to y \quad (2)$$
$$not(0) \to 0 \quad (3)$$
$$y + not(y) \to 0 \quad (4)$$
$$not(y) + y \to 0 \quad (5)$$
$$y \,\&\, not(y) \to 0 \quad (6)$$
$$not(y) \,\&\, y \to 0 \quad (7)$$
$$f(x) \to 0 \quad (8)$$
$$1 \to 0. \quad (9)$$

One could foresee this result, since the roles of 0 and 1 are reversed in the completion: we choose $1 > 0$ instead of $0 > 1$.

Let us observe a third computation. It is interesting to observe that a different ordering can lead to different rules during the completion, but gives the same resulting system as in the first completion.

Start the completion as in the first case until the second precedence choice proposition. Let us here choose the second unequality: $0 > not$. The rule

$$not(not(1)) \to 1$$

Precedence		Performances						
0>1, not>0, f>1	A	8	2.7	0.71	1.56	3.84	27	15
0>1, 0>not, not>1, f>1	A	5	1.09	0.49	1.43	2.84	20	12
0>1, 0>not, +>1, f>1, not>1	A	6	1.14	0.59	2.75	1.98	20	12
0>1, 0>not, +>1, notf>1, &>1	A	6	1.18	0.59	2.09	2.62	20	12
not>1, 1>0, f>1	B	8	2.57	0.76	1.67	3.94	27	15
not>1, not>0, f>1, 0>1	A	9	2.87	0.85	2.39	3.70	27	15
not>1, not>0, f>1, 1>0	B	8	2.31	1.02	1.94	3.46	27	15
not>1, 0>not, f>1	A	6	1.79	0.58	1.77	2.67	24	14
1>0, 1>not, not>0, f>0	B	5	1.07	0.53	1.88	1.92	19	11
1>0, 1>not, 1>0, &>0, f>0, not>0	B	6	1.65	0.55	1.87	1.98	19	11
1>0, 1>not, &>0, f>0, +>0	B	5	1.04	0.51	2.07	2.08	19	11

We mention respectively: obtained system, total runtime, unification, rewriting, ordering, overhead time, number of critical pairs and number of ordered equations. Times are given in seconds.

Let us remark that two identical orderings can be obtained by completing precedence in a different order, with different completion processes (for example, the first and sixth cases below).

References

1. N.Dershowitz, "Termination", Proc. of 1rst Conf. on Rewriting Techniques and Applications, Edited in LNCS, Springer 1985

2. N.Dershowitz & L.Marcus, "Existence and Construction of Rewrite Systems", Research Report University of Illinois, 1982

3. R.Forgaard, "A Program for generating and analyzing Term Rewriting systems", Master of Science in Computer Science, MIT, 1984

4. I.Gnaedig "Trois extensions du processus d'orientation des regles de reecriture dans REVE", Research Report CRIN 83-R-097, Nancy, France, 1983

5. D.Knuth & P.Bendix, "Simple Word Problems in Universal Algebras", Computational Problems in Abstract Algebra, Ed. Leech J., Pergamon Press, 1970

6. P.Lescanne, "Term Rewriting Systems and Algebra", Proc. of CADE 7, Napa Valley, California, Edited in LNCS, Springer 1984

7. P.Lescanne, "Divergence of the Knuth-Bendix Completion Procedure and Termination Orderings", Bulletin of EATCS, Number 30, October 1986

8. J.Meseguer & J.Goguen, "Deduction with Many-Sorted Rewrite Rules", SRI International Technical Report, Menlo Park, California, USA

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique